開発環境でのソフトウェアの構成

この文書では開発環境を構築するための手順をまとめ、
開発を行うためのツールを紹介する。

Windows での開発環境の構築

InstantRails

InstantRails を公式サイトから取得しインストールする:
http://instantrails.rubyforge.org/wiki/wiki.pl?Instant_Rails

* rails_app 以下に rfw を置く。
* config/database.yml を作成し、それに対応するデータベースを
「Configure - Database (via PhpMyAdmin)」などから作成する。
* もしくは database.yml.mysql.db_setup を database.yml としてコピーして、
lib/tasks/db_setup.rake を使って、次のコンソールで「rake db_setup」と
「rake RAILS_ENV=test db_setup」と「rake RAILS_ENV=production db_setup」で作成する。

* 「Rails Applications - Open Ruby Console Window」を開く。
* gem install gettext で Ruby-GetText-Package の gem をインストールする。
(選択肢は「(mswin32)」とついているものを選ぶ。複数の選択肢が該当する場合には
その中でバージョンの新しいものを選ぶ。)
* gem install mongrel -y で mongrel をインストールする。
(script/server で WEBrick の代わりに mongrel が使われるようになる。)
* gem install ap4r -y で ap4r をインストールする。
* BackgrounDRb が依存している gem install slave -y で slave をインストールする。
(もう一つの依存の daemons は mongrel が依存しているので入っているはず。)
* gem update -y で rails を最新安定版に更新する。
* cd rfw で rfw のディレクトリに入り、
* 「rake db:migrate」でテーブルを作成する。
* 「rake db:fixtures:load」でテーブルにデータを読み込む。
* 「rake makemo」で mo ファイルを作成する。
* 「Rails Applications - Manage Rails Applications...」を開き、
* rfw にチェックを入れ、
* 「Start with Mongrel」を押す。
* 「** Use CTRL-C to stop.」まで表示されたらブラウザで http://localhost:3000/ を開く。

gted :: GetText EDitor (オプション)

http://gted.sourceforge.net/
* ヘルプ - ソフトウェア更新 - 検索およびインストール
*「インストールする新規フィーチャーを検索」を選んで次へ
*「新規リモートサイト」で URL に「http://gted.sourceforge.net/update」
を入力して名前は適当に入力してOK
* gted にチェックを入れて終了
* 検索結果から gted にチェックを入れて次へで進んでインストールする。

行番号が「-」に対応していないようで、Ruby-GetText-Packageが
データベースから取り出してくる行以降がEntriesで見えない。

Oracle (オプション)

* あらかじめデータベースとユーザを作成する。
* データベースは development、test、production の3つを用意する。
* config/database.yml.oracle を config/database.yml にコピーし、
* development、test、production それぞれについて項目 database、
username、password を作成したものと一致するよう変更する。
(項目 database を空白にすることで既定値を利用できる場合もある。)

Copyright(c) 1996-2008 Earth Inter Systems Co.,ltd. All Rights Reserved.

データベースのバージョンアップ

* 「rake VERSION=0 db:migrate」で一旦テーブルを消す
* データベースの構造を変えている場合があるので
* 「rake db:migrate」でテーブルを作成し直し
* 「rake db:fixtures:load」でテーブルにデータを読み込む。

ローカライズメッセージの追加

* ローカライズ対象となる文字列に複数形による変化がない場合は、
  s_("...") というようにs_() で囲む。複数形による変化がある場合は ns_() で囲む。
* 「rake updatepo」でpoファイルを更新
* po/ja/rfw.poを編集
* 「rake makemo」でmoファイルを更新
* script/backgroundrb の中でパスを決め打ちになっているので、
  vendor/plugins/backgroundrb という名前で入れておかないと動かない。
* 2.1 では rake rake backgroundrb:stop と rake backgroundrb:restart が
  使えないので、script/backgroundrb stop などを使う。
* 開発環境では「script/backgroundrb run -- -l」で起動するとscript/server のように Ctrl+C で停止できる。
* 「--」の前は daemons の引数で、後ろは backgroundrb の引数。
* 自動で再読込はしないので、lib/workers 以下のファイルを含めて
  依存しているファイルを変更した場合は再起動する必要がある。
* デーモンにする場合は script/backgroundrb start で起動する。
* script/backgroundrb console で MiddleMan のコンテキストで irb を実行できる。
* 詳細は vendor/plugins/backgroundrb/README を参照。

偽メールサーバ

* 「ruby script/fake-smtpd.rb」のように起動する。
* $KCODE にあわせて自動で文字コードを変換しているので、Windows では
  ruby -Ks script/fake-smtpd.rbで起動すれば文字化けしない。
* config/environments/development.rb で 127.0.0.1 の 10025 に送信するようにしているので、
  development 環境で ap4r と同時に起動しておけばメール送信回りの動作確認に使用できる。

非同期でのメール送信について

* BackgrounDRb のスケジュール機能を利用して非同期にメールを送信することができる。
* メール履歴を確認して送信に失敗しているメールを再送するために使う。
* 手順: BackgrounDRb および メールサーバを起動してから script/schedule_to_deliver を実行する。

注意

- Rails を起動する場合には偽メールサーバを立ち上げておく。
  さもないとメールの送信を送る際にエラーが生じる。

Linux での開発環境の構築

ディストリビューションに Debian etch を利用している場合は、
root 権限で次のコマンドを実行してインストールする:

```
# aptitude install sqlite3 postgresql mysql emacs iceweasel ruby rubygems
# gem install sqlite3-ruby rake rails gettext mongrel piston ap4r capistrano
```

IDE を利用する場合には適宜公式サイトからインストーラーを取得する。

Copyright(c) 1996-2008 Earth Inter Systems Co.,ltd. All Rights Reserved.

postgresql-8.1 (オプション)

* ユーザ作成
* sudo -u postgres createuser -S -d -R -P rfw
* データベース作成 (testは自動的に作成される)
* sudo -u postgres createdb -O rfw rfw_development
* sudo -u postgres createdb -O rfw rfw_production
* config/database.yml.postgresql を config/database.yml にコピーして
  password を設定し、host: localhost のコメントアウトを外し、
  testと production のところにもコピーする。

注意

- 基本的には Windows での開発環境のセットアップと同様にする。

紹介

IDE

:Aptana Radrails
  http://www.aptana.com/rails/
  Eclipse ベースの Aptana 統合開発環境で動作する高機能なプラグイン。
  - アウトラインエディタ
  - ブラウザ
  - ソースコードのテンプレート
  - デバグコンソール
  などを GUI で利用できる。

:NetBeans 6.1 IDE (予定)
  http://ja.netbeans.org/
  次期リリースのアドオンで Ruby on Rails をサポートする。
  - コードアシスト機能
  - シンタックスハイライト
  - アウトライン機能
  などを強化している。

rake

フレームワークは Ruby on Rails 全体を開発環境として利用する。
特に rake は他の IDE のベースになるユーティリティである。
複数のタスクから構成され、コマンドの引数でタスクを指定する。
良く利用するタスクを以下に挙げる:

- データベースの migration
      rake db:migrate
- データベースへの初期データ(fixtures)のロード
      rake db:fixtures:load
- rdoc ドキュメントの再構築
      rake doc:reapp
- テスト
      rake test
- タスクの一覧
      rake --tasks

Selenium (Selenium on Rails)

Web アプリケーションとしての UI をテストするためのツール。
クロスブラウザ対応であり、Rails と統合されたプラグインで動作する。

Copyright(c) 1996-2008 Earth Inter Systems Co.,ltd. All Rights Reserved.